

ROLE OF IN-CIRCUIT EMULATOR IN ONBOARD DSPs FOR PERFORMANCE EVALUATION

Noureen Owais*,
Quality Assurance Directorate, SUPARCO
* noureenowais@yahoo.com

Abstract-The ADSP-21065L of Analog Devices is an extraordinary member of family of 32-bit processors, containing both fixed and floating point capabilities. In this research, a customized developed board containing the above processor was used and its behavior with its dedicated emulator was studied by devising certain experiments. This work will provide a guideline for those who wish to use this customized board for their application. Moreover the behavior what we have studied during our experimentation is hard to find in any guide.

Keywords: D-module, , VDSP++ (Visual DSP), Boundary scan , JTAG (Joint Test Action Group), target board , IEEE 1149.1.

I. INTRODUCTION

EMULATORS are actually the combination of hardware (prototype testing) and software (Simulator) that examine the program operation/execution on actual hardware. These are dedicated hardware for the said combination of hardware and software.

Work in this area depends on customized target boards that's why anyone who utilizes this make its own set up and gather its own database for any particular application. What we are representing here is specific target board, so database and experimental results are not easily available and they are the intellectual property of any organization.

This paper discusses in detail the function of emulator and specifically Analog Devices Inc. EZ ICE Emulator for Digital Signal Processors (ADSP-21065L) [1].

During whole experimentation some new problems are discovered which will be discussed later in this research.

This paper will cover the following aspects: brief architectural detail of ADSP-21065L, EZ-LAB development architecture, JTAG connector, annunciations on emulator, IEEE 1149.1, Emulator power sequencing, Target board (D.signT :D-Module 21065L) [2], experimental setup(block diagram), D-module adaptor board, power requirement of D-module, development of interfacing cable, event of loading up VDSP when all components are interfaced, experiments regarding step by step checking/scanning the loop, understanding and learning Visual DSP and debugging and building up a project (code development) in VDSP.

II. A BRIEF ARCHITECTURAL DETAIL OF ADSP 21065L

Basically, architecture refers to those characteristics of a system that directly impact the logical execution of a program. ADSP-21065L architectural features are as follows [1]:

A. Computation Units: Parallel & Independent

The three units which perform parallel computation performing single-cycle instructions are: Arithmetic/logic unit (ALU), multiplier and shifter. For maximizing computational throughput these three units are arranged in parallel,. These computational units supports the following data formats: IEEE 32-bit single precision floating point, extended precision 40-bit floating point and 32-bit fixed point.

B. Instruction Of Single Cycle Fetch

Program memory and data memory are two memory spaces which are separately classified in Harvard memory architecture. Two separate bus sets of the processor core access to these memory spaces by, two simultaneous accesses to memory. This arrangement two folds the memory bandwidth. The processor can concurrently fetch two operands and an instruction with separate program and data memory buses, all in one cycle.

C. Instruction Set: Flexible

The instruction word of 48-bit accommodates diverse of parallel operations, for succinct programming. In a single instruction the ADSP-21065L can perform a multiply, an add, a subtract and a branch.

D. Serial Port Organization in ADSP-21065L

Two synchronous serial ports in the ADSP-21065L provide an interface to digital and mixed signal peripheral devices. Independent transmit and receive channels of each port provide flexibility for serial communication.

E. General Purpose Input/output Programmable Ports and Timers

This processor has two independent timer blocks. It has also twelve general purpose I/O pins (programmable) that functions either as input or output as programmed.

F. Development Tools

The ADSP-21065L is augmented with development tools (both in hardware and software). EZ-ICE and Visual DSP integrated development environment fully support ADSP-21065L. For control and examination of the target processor during emulation, Emulator uses the IEEE1149.1 JTAG test access port of ADSP-21065L. Using EZ-ICE inspection and modification of memory, registers and processor stacks can be done. Target system loading or timing issues are not affected by the emulator.

G. Memory Architecture

The memory on the 21065L can be configured as a maximum of 16K words comprising of 32-bit data, 34K words of 16-bit data, 10K words of 48-bit instructions or combination of varied word sizes up to 544Kbits. The memory can be accessed as 16-bit, 32-bit or 48-bit. Every memory block can store combination of program code and data. For most efficient accesses one block which stores data, use the Data Memory (DM) bus for transfers, and the other block stores instructions and data, use the Program Memory (PM) bus for transfers. This will assure single-cycle execution with two data transfers while using the Data Memory and Program Memory buses with one bus dedicated to each memory block.

III. EZ ICE Emulator Architecture

A. JTAG Connector(Emulator Port)

A 14-pin JTAG emulator header is used to interface with the DSP. Pin 3 of the JTAG header is pulled out to prevent unintentional placing of the pod with the target backwards [3] [4].

B. Input/output Characteristics of JTAG Emulator Pod

The JTAG pod is capable of bearing up to 5VDC. It will operate with all Analog Devices JTAG family DSPs with I/O voltages of 5V, 3.3V, 2.5V and 1.8V. The pod will drive 5V targets with 3.3V logic levels, which comply the 5V logic level .

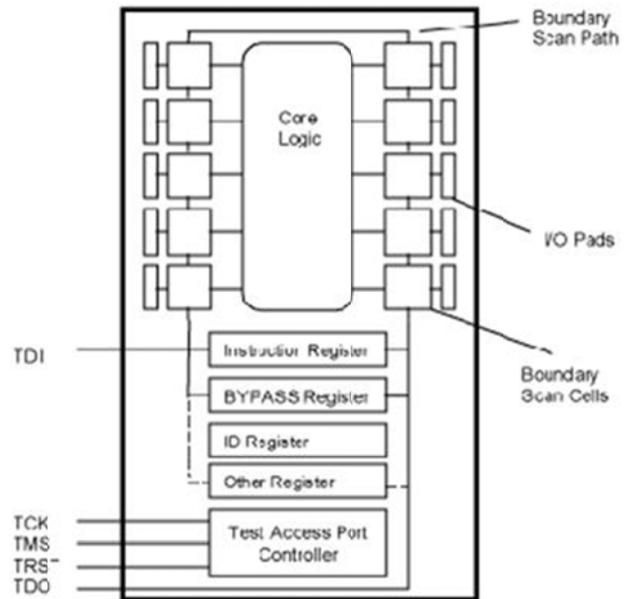
C. Boundary Scan using IEEE 1149.1)

Here is a brief introduction to the IEEE (JTAG) specification and the concept of Boundary Scan. The Joint Test Action Group (JTAG) developed the IEEE Std. 1149.1 [3] standard utilizing Boundary-scan for testing connections on printed circuit boards at the device pin level.

D. Boundary Scan

Boundary scan provides a method for testing interconnects on a PCB without having to use physical probes. By using internal boundary scan cell with multiplexing and latching capabilities, data can be shifted in and out of device interconnects in a serial format. Each boundary scan cell using its own multiplexer and latch logic is connected to a device pin, and all boundary scan cells are connected forming a serial chain.

The chain will include the boundary scan cells for all or some of a device’s pins (user defined) which is connected to other devices (user defined) on the PCB.



The above figure shows a diagram of how a part implementing IEEE Std. 1149.1 would connect the individual boundary scan cells, and the JTAG controller logic.

E. Annunciations on JTAG Emulator

The four LEDs located on the emulator has following description [4]:

- 1) 1.8V LED–This LED indicates that the ICE will drive all signals at 1.8V compliant levels.
- 2) 2.5V LED–This indicates that the ICE will drive all signals at 2.5V compliant levels.

- 3) 3.3/5V LED–This will indicate that the ICE will drive all signals at 3.3V compliant levels.
- 4) All 3V LEDs lit show that the voltage has not been set correctly.
- 5) ENABLE /POWER LED–This will turn on when a VisualDSP++ session is running. This LED shuts off each and every time the VisualDSP++ session is closed. When ENABLE /POWER LED is OFF, it indicates all outputs of the POD logic connected to the target are tri-stated. During the target power up sequencing tri-stating the outputs of the pod will prevent the target processor from entering an unknown state when VDSP++ is inactive. If the Emulator has an ‘Enable/Power’ LED, this should be green when power is applied. It should be amber when connected to a session or the ICE Test utility is being used.

F. JTAG Emulator Power Sequence

1. Prior to attaching it to a target board JTAG Emulator should be powered up, and connected to a host PC.
2. the emulation software is not started or executing when power is not applied to the target, in this way emulation errors can be avoided.
3. It must be ensured that the emulation software (VDSP++) is not started or executing, when attaching the JTAG emulator to a target with or without power,.
4. If emulator has an ‘Enable/Power’ LED, this should be *green* when power is applied. It should be *amber* when connected to a session or the ICE Test utility is being used.

G. SHARC Emulator Configuration

The ICEPAC software is an embedded version of the EZ-ICE in circuit emulator which is a development tool for real time debugging. The ICEPAC is inserted in the JTAG scan path through an ICEPAC interface. The emulator software always checks in the initialization file, WICE060.ini, for the JTAG chain (target systems) information [5]. During testing of emulator through VDSP++, following tests are performed [6] :

- 1) Opening Emulator Interface.
- 2) Resetting ICEPAC module.
- 3) Testing ICEPAC memory.
- 4) Determining scan path length.
- 5) Performing scan test.
- 6) Continuous scan mode.
- 7) Debug mode.

1) Opening Emulator Interface:

Opens the emulator driver (if required by the operating system) and thus turns ON the emulator pod.

If this test fails, the most common cause is that the emulator driver is not installed properly.

2) Resetting ICEPAC module:

Resets the JTAG controller on the emulator hardware and checks for proper reset conditions.

If this test fails, there is likely a problem with the ICE test utility itself.

3) Testing ICEPAC memory:

Tests the memory on the JTAG controller on the emulator hardware and verifies that all locations can be read and written to. It also resets the JTAG port of each JTAG device located on the target hardware.

Generally if this test fails continuously, the emulator is faulty.

4) Determining scan path length:

Determining the number of different JTAG devices located on the target hardware to which the emulator is connected. This is accomplished by placing all the devices in the BYPASS mode and scanning a single ‘1’ through the chain.

If this test fails, the ICE may be faulty, but can also indicate a problem with the target.

5) Performing scan test:

Tests the integrity of the scan path on the target hardware using each device’s BYPASS register. Each time the test runs, 25 packets of 256 bytes are shifted through the BYPASS registers of the all the JTAG devices in the scan path. The data shifted in is compared to the data received. The numbers of bytes are displayed.

If mismatches are found, an error is reported and the test fails.

6) Continuous Scan Mode:

When the continuous scan option is selected and a test is started by clicking the ICE test dialog box’s start button, the test runs normally to the performing Scan test item. The test loops continuously until the test is stopped by clicking the dialog box’s stop button. This scan mode sends a very large amount of data compared with all previous tests.

If this mode fails it usually means there is noise on the JTAG signals. They should be observed using a scope to see if there is excessive ringing or noise.

7) Debug Mode:

When the Debug mode option is checked and a test is started by clicking the ICE test dialog box’s start button, the test will halt before each highlighted test to allow setting logic analyzer for capturing the JTAG scan signal.

IV. Understanding and Learning Visual DSP++

Software plays a vital role in the design of embedded systems design [7] .

Before Building and debugging any project in VDSP++, one must be [8]:

- Fluent in working with VDSP,
- compliant of the front end for all targets and platforms,
- Understanding of how program sections and memory sections relates to physical memory,
- Be able to access peripherals.
-

A. Visual DSP++ Features:
-code development tools:

Following are the main tools: assembler, splitter, C/C++ compiler, loader and linker. Dialog boxes are used to specify options for these tools by instead of complicated command-line scripts.

B. project build options.

VisualDSP++ facilitates to build files or projects selectively, and update project dependencies. Control builds at the file or project level.

C. movement between debug and build activities.

there is free movement between editing, building, and debugging activities after starting the debug session.

D. manifold language support.

Multiple language support is available in such a way that debug programs can be written in C, C++, or assembly, and program can be viewed in machine code.

E. development Tools

Code development tools include:

- 1) Assembler
- 2) C/C++ compiler.
- 3) Linker
- 4) Run time library, DSP and C-run time library routines.
- 5) Splitter
- 6) Loader
- 7) Emulator
- 8) Simulator

F. Connecting to a Debug Session

When the main VDSP++ window opens it will not connect to any session when it is loaded first time.

Simulator is a software model of the processor. Simulators offer a unique insight to the internal workings of the processor (pipelines, caches, and more), which is not possible with hardware-based sessions. Simulators offer unique advantages; i.e. that no external hardware is required. The downside is that a simulator is several orders of magnitude slower than actual hardware.

G. Building and Running a C program

Preprocessing, assembling, and linking on projects and files is termed as build. During a build phase, VisualDSP++ processes project files that have been modified as well as project files that include modified files.

There exists a step by step procedure for building and running a C program in VDSP++ environment.

From the available sessions debug session is chosen and from there project is loaded.

Then project is build, once the build is successful it can be loaded in the processor.

H. Code development in VDSP++ environment

Program development includes the following steps in the Visual DSP++ environment, [8]:

- a. A project is created.
- b. Project options are configured.
- c. Project source files are added and edited.
- d. Project build options are specified.
- e. A debug version is built i.e. executable files of the project.
- f. A debug session is created and the executable is loaded.
- g. The program is run and debugged.
- h. A release version of the program is built.

V. Experimental Set Up for Configuring Emulator in the Circuit (Block Diagram)

In this experiment a customized embedded module (D-module) containing ADSP-21065L is used, in which it is showed how emulator behavior varies with any particular application. Despite the versatility of D-module constraints may exists which prevent their use in certain application [2].

The Experimental setup consists of following elements:

- 1) DC Power Supply.
- 2) 220 V AC Power Supply.
- 3) Personal Computer(PC)
- 4) EZ-ICE Emulator.
- 5) D-Module (ADSP-21065L).
- 6) D-Module Adapter.
- 7) USB Port.

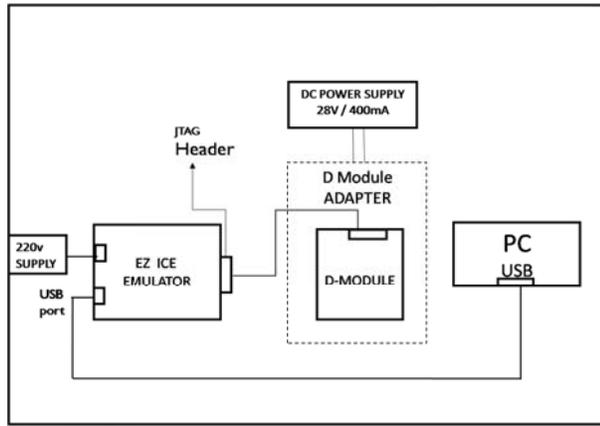


Fig 1: Block diagram of the experimental setup

A. *Interfacing Components of the Experimental Setup*

Following are the various interfaces present in our devised experimental setup:

- 1) Power requirement of D Module adapter board.
- 2) Interfacing of D Module with adapter board.
- 3) Power requirement of EZ ICE Emulator.
- 4) Connecting Emulator to the target board (D Module).
- 5) Connecting Emulator to the PC.
- 6) Specific requirement of JTAG connector.
- 7) Interfacing Visual DSP with hardware.

B. *Development of Interfacing Cable*

The Interfacing cable is required to be developed in order to fulfill the requirement of interfacing various elements of the set up and more specifically for powering up the D-Module Adapter Board.

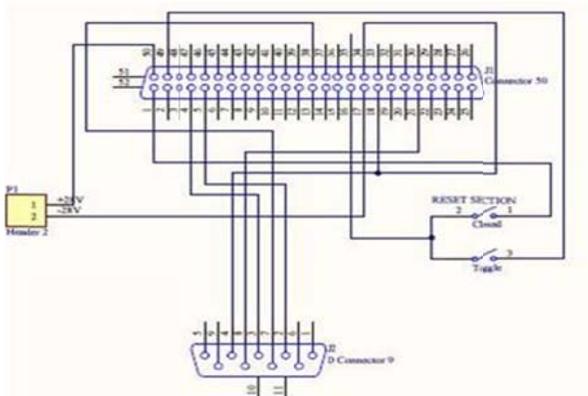


Fig 2: Interfacing cable layout

C. *Target Board (D-Module 21065L)*

Hardware and software are integral unit in any real-time system. This is the case with DSP system, therefore the D-module hardware and software has been finely tuned to give

optimum performance. The D.Module (DSP based computer module) offers a standardized hardware platform for embedded DSP applications.

The D.Module.BIOS functions are the built-in feature of the module permanently stored in the Flash Memory and booted at power-up or reset. All peripheral devices' initialization and programming is handled by it.

Advantage of using this board is that it has bus architecture which allows on-board emulation to take over control via one of the bus. This allows emulation to take full control over all of the processor buses.

The firmware also includes a Setup Utility.

This customized board (D-Module) has following devices:

- 1) ADSP-21065L: Core Processor.
- 2) ADM-3202: Line Driver.
- 3) ADP-3310: Voltage regulator Controller.
- 4) 4016: SRAM.
- 5) SC16C650B: UART.
- 6) XCR 3032XL: CPLD.

For datasheets of the above peripherals refer [9], [10], [11],[12],[13].

D. *Experiments carried Out for scanning/ checking the loop.*

Experiment #1: EZ-ICE is disconnected from the target board; powered up and then ICE Test utility runs.

Result:

The following tests passed:

- 1) Opening Emulator Interface.
- 2) Resetting ICEPAC module.
- 3) Testing ICEPAC memory.

While the following tests failed:

- 1) Determining scan path length.
- 2) Performing scan test.

Due to the reason that JTAG emulator is not connected to the target, so as to perform scan test.

Observations:

When power of emulator is reset it did not pass; and when checked in device manager of the system, device drivers automatically uninstalled. Reason of this still unknown

Conclusion:

When power of emulator is reset, it did not pass, although all power sequencing steps were followed. After thorough debugging it was found that device drivers were automatically uninstalled. This has been tested several times and the reason for this un installation is still unknown. The results of this experiment shows that this

emulator has reflected this changed behavior on this dedicated device, so anyone who wants to use this customized embedded board for particular application must be aware of this unknown behavior.

Experiment #2: EZ-ICE is connected to the target board when connection is hot.

Result:

Following tests passed:

- 1) Opening Emulator interface.
- 2) Resetting ICEPAC module.
- 3) Testing ICEPAC memory.
- 4) Determining scan path (With continuous scan & with debug mode).
- 5) Performing scan test.

Observations:

1) *While performing simple ICE Test:*

All tests are passed.

2) *While performing continuous scan:*

All tests are passed and scan test of the target was performed. Stop button appeared in place of start button.

3) *While performing debug mode ICE Test:*

All tests are passed, pop up window appeared allowing user to debug each test. While performing any test every time debug mode window gives user a permission to debug the test.

4) Conclusion:

When all the connections are hot the entire set of tests passed; with this configuration successful results could be achieved.

VI. Conclusion

For performance evaluation of on-board DSPs two experiments were performed which were described in this paper which showed that for any particular application dedicated emulator for any processor may exhibit changed behavior to some extent. Practically in a system in which this board had been used, several problems were encountered among which we have tested and discussed only power issues in our experimentation. Although literature available has laid down power sequencing steps but still practical results varied. So this work may help the beginners/researchers those who intend to use this hardware (combinational hardware) for their desired application.

VII. Future Work

There are many areas which can be further explored for performance evaluation

Acknowledgments

This research was carried out in the Electronics Section (QA Directorate) of SUPARCO. The author would like to express gratitude Mr. Ahmed Bilal (Chairman SUPARCO) for provision and approval of facilities. Further acknowledgments to Dr. Sajid Mirza (Senior Chief Manager) for his support, Dr. Rasheed Ahmed Baloch (Deputy Chief Manager, Director Quality Assurance), and Mr. Atiq Ur Rehman (GM, LEO Satellite Center) for their valuable suggestions and guidance throughout this work and Ms. Aneela Perveen being part of the team for her technical assistance.

References

- [1]. "Analog Devices DSP Micro Controller ADSP-21065L Data sheet", Analog Devices, Rev.C, 2003, pp.1-44.
- [2]. "Design T: D-Module (21065L)", (2011, December 12). Available: <http://www.dsignt.de>
- [3]. David M. Doyle, "EE-68 Analog Devices JTAG Emulation Technical Reference", Analog Devices, Rev.10, 2008, pp. 1-20.
- [4]. "HPUSB, USB and HPPCI Emulators Users Guide", Analog Devices, Rev.3.1, 2009, pp.1-41.
- [5]. "EE-43: Configuring Emulator Software for your Target", Analog Devices, 1998, pp. 1-5.
- [6]. Colin Martin, et al, "EE-175: Emulator and EZ-KIT Lite® Evaluation System Troubleshooting Guide", Analog Devices, Rev.10, 2007, pp. 1-12.
- [7]. Gert Goossens, et al, "Embedded Software in Real-Time Signal Processing Systems: Design Technologies" Proceedings of the IEEE, vol. 85(3), 1997, pp. 1-19.
- [8]. "VISUAL DSP++5.0, Getting Started Guide", Analog Devices, Rev.3, 2007, pp.1-128.
- [9]. "Analog Devices, ADM3202/ADM3222/ADM1385 Data sheet", Analog Devices, Rev.E, 2011, pp.1-16.
- [10]. "Analog Devices, ADP3310 Data sheet", Analog Devices, Rev.A, 1999, pp.1-8.
- [11]. "Analog Devices, CMOS SRAM K6R4016V1D Data Sheet", Analog Devices, Rev.0.3, 2001, pp.1-12.
- [12]. "SC16C650B UART Data Sheet", NXP Phillips, Rev.4, 2009, pp.1-48.
- [13]. "Cool Runner XPLA3 CPLD, DS012 Data Sheet", XILINX, vol. 2.5, 2009, pp.1-12.